

JP2003188737(A)

INTERLEAVE PROCESSING METHOD AND INTERLEAVE PROCESSOR

Publication number : **2003-188737**
Date of publication of application : **04.07.2003**

Int.Cl. **H03M 13/27**
G06F 11/10
H03M 13/29

Application number : **2001-380759** Applicant : **SONY CORP**
Date of filing : **13.12.2001** Inventor : **UCHIDA MASAKI**

Abstract:

PROBLEM TO BE SOLVED: To surely perform rearranging processing with a little hardware quantity by reducing a memory required for an interleaver to be used for a turbo- code or a LDPC code.

SOLUTION: The interleave processor is composed of a rearrangement table TB written with rearrangement data for assigning a non-interleaved data symbol sequence or data bit sequence to a prescribed block while defining a partial sequence dividing an interleaved sequence for each of the prescribed number of symbols or the prescribed number of bits relatively to an inputted data symbol sequence or a data bit sequence as a block, a selector SL for assigning the non-interleaved data symbol sequence or the data bit sequence to the prescribed block according to the rearrangement data applied by the rearrangement table TB and a plurality of memory blocks MB1-MB64 to be assigned with the data symbol sequence or the data bit sequence via the selector SL.

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2003-188737
(P2003-188737A)

(43)公開日 平成15年7月4日(2003.7.4)

(51)Int.Cl.
 H 03 M 13/27
 G 06 F 11/10
 H 03 M 13/29

識別記号
 3 3 0

F I
 H 03 M 13/27
 G 06 F 11/10
 H 03 M 13/29

テ-マ-^ト (参考)
 5 B 0 0 1
 3 3 0 N 5 J 0 6 5

審査請求 未請求 請求項の数 6 ○ L (全 7 頁)

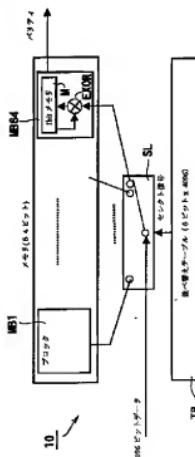
(21)出願番号	特願2001-380759(P2001-380759)	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川6丁目7番35号
(22)出願日	平成13年12月13日(2001.12.13)	(72)発明者	内田 雅貴 東京都品川区北品川6丁目7番35号ソニー 株式会社内
(23)代理人		(74)代理人	100067736 弁理士 小池 晃 (外2名)
(24)請求項		F ターム(参考)	5B001 AA10 AC05 AD06 AE04 AE07 5J065 AA01 AA03 AB01 AC02 AD10 AE06 AF03 AG06 AH06 AI09

(54)【発明の名称】 インターリープ処理方法及びインターリープ処理装置

(57)【要約】

【課題】 ターボ符号あるいはLDPC符号などで用いられるインターリーバにおいて必要とされるメモリを削減し、少ないハードウェア量にて確実に並べ替え処を行ふ。

【解決手段】 入力されるデータシンボル系列あるいはデータビット系列に対し、インターリープ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックとして、インターリープ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるための並べ替えデータが書き込まれた並べ替えテーブルTBと、上記並べ替えテーブルYBにより与えられる並べ替えデータに従って、インターリープ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるセレクタSLと、上記セレクタSLを介して上記データシンボル系列あるいはデータビット系列が割り当てられる複数のメモリブロックMB1～MB64とからなる。



【特許請求の範囲】

【請求項1】 入力されるデータシンボル系列あるいはデータビット系列に対し、インターリーブ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックとして、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる操作を行うことにより、入力される系列に対するインターリーブ操作の代わりとすることを特徴とするインターリーブ処理方法。

【請求項2】 インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる際に、過去に当該ブロックに割り当てられたデータ及び現在割り当てられようとしているデータを引数とするブロック単位で実施される所定の演算を行うことを特徴とする請求項1記載のインターリーブ処理方法。

【請求項3】 ターボ符号あるいはこれに類する符号にて用いられるブロック符号において、ブロック符号の符号長を上記ブロックとし、ブロック符号の制約条件を上記ブロックに割り当てる演算を行うことを特徴とする請求項1記載のインターリーブ処理方法。

【請求項4】 入力されるデータシンボル系列あるいはデータビット系列に対し、インターリーブ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックとして、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるための並べ替えデータが書き込まれた並べ替えテーブルと、上記並べ替えテーブルにより与えられる並べ替えデータに従って、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるセレクタと、上記セレクタを介して上記データシンボル系列あるいはデータビット系列が割り当てられる複数のメモリブロックとからなることを特徴とするインターリーブ処理装置。

【請求項5】 上記複数のメモリブロックは、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる際に、過去に当該ブロックに割り当てられたデータ及び現在割り当てられようとしているデータを引数とするブロック単位で実施される所定の演算を行う演算手段をそれぞれ備えることを特徴とする請求項4記載のインターリーブ処理装置。

【請求項6】 ターボ符号あるいはこれに類する符号にて用いられるブロック符号において、ブロック符号の符号長を上記ブロックとし、ブロック符号の制約条件を上記ブロックに割り当てる演算を上記演算手段にて行うことと特徴とする請求項5記載のインターリーブ処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、並列連接量込み(Turbo)符号あるいは、これに類する符号に用いられるインターリーブを用いパリティ分散伝送を行っている信号処理装置の符号化部及び復号化部に含まれるインターリーブに適用されるインターリーブ処理方法及びインターリーブ処理装置に関する。

【0002】

【従来の技術】 近年、例えば、移動体通信や深宇宙通信といった通信分野、地上波又は衛星デジタル放送といった放送分野、記録媒体を介して情報の記録／再生を行う情報記録分野の研究が急速に進められているが、それに伴い、誤り訂正符号化及び符号の高効率化を目的として符号理論の研究も盛んに行われている。

【0003】 誤り訂正符号の性能は、復号誤り率、符号化率、計算量などで評価され、その代表的論理的限界として、所謂シャノンの通信路符号化定理により与えられるシャノン限界が知られている。

【0004】 このシャノン限界に近い性質を示す符号として、並列連接量込み(Turbo)符号や低密度パリティ検査(LDPC: Low Density Parity Check)符号が注目されている。

【0005】 これらの符号はターボ符号やLDPC符号は、複数の量込み符号化器とインターリーバとを組み合わせて構成される符号化装置により生成される。そして、復号側では、複数の軟出力(soft-output)を出力する復号回路の間で入力データに関する情報を授受することにより最終的な復号結果を得ることができる。

【0006】 ターボ符号の符号器と復号器の基本構成について、図3、図4を参照して説明する。

【0007】 図3に示すようにターボ符号の符号器100では、入力データ系列が第1のエンコーダ101に入力され、この第1のエンコーダ101にて第1のパリティビット列P1が作られる。データ系列は、同時にインターリーバ102で並び替えられて、第2のエンコーダ103に入力され、この第2のエンコーダ103にて第2のパリティビット列P2が作られる。第1及び第2のパリティビット列P1、P2は、第1の多重化器104にて間引きしながら多重化され、さらに第2の多重化器5にて入力データ系列と多重化され通信路に送出される。

【0008】 また、ターボ符号の符号器200は、図4に示すように、第1のデコーダ201、インターリーバ202、第2のデコーダ203及びデインタリーバ204から構成される。

【0009】 第1のデコーダ201は、入力された受信データ系列に復号処理を施し、各種情報シンボルの復号結果とその信頃度情報を出力する。インターリーバ202は、入力された受信データ系列と上記第1のデコーダ201により得られた信頃度情報を並び替え処理を施

す。第2のデコーダ203は、上記第1のデコーダ201から上記インターリーバ202を介して供給される信頼度情報と受信データ系列用いて復号処理を行い信頼度情報を計算し、デインターリーバ204を介して第1のデコーダ201に送る。

【0010】2回目以降の繰り返しは、第1のデコーダ201は第2のデコーダ203からの信頼度情報と受信データ系列用いて復号処理を実行する。これを数回から十数回繰り返した後、最終判定を行い最終的な復号結果として出力する。

【0011】復号器200におけるインターリーバ202の並べ替えは符号器100と同じである。復号器200では、信頼度情報系列に加えて受信データ系列も並べ替える。これは、パリティビット列の順に信頼度情報系列と受信データ系列を並べ、第2のデコーダ203での処理の順に合わせるためにある。デインターリーバ204は、インターリーバ202の並べ替えを元に戻す処理を行う。

【0012】ここで、従来の、例えばターボ符号あるいはLDPC符号などで用いられるインターリーバは、入力データ系列を並べ替える操作のみを行っている。4096ビットのデータ系列に対して並べ替えを行うインターリーバ310と64/65シングルパリティ符号回路320の構成例を図5に示す。

【0013】入力データ系列は、インターリーバ310において、4096ビットごとに区切られ、並べ替えテーブル311のデータにしたがって動作するセレクタ312により4096ビットごとに並べ替え処理が実施され、4096ビットのメモリ313に格納される。

【0014】64/65シングルパリティ符号回路320は、排他的論理と回路321と、この排他的論理と回路321の出力をラッチするラッチ回路322からなり、インターリーバ310により並べ替えられた入力データ系列について、上記排他的論理と回路321に入力データ系列と上記ラッチ回路322のラッチ出力との排他的論理と取ることにより、64/65シングルパリティ符号を生成する。

【0015】

【発明が解決しようとする課題】ところで、ターボ符号あるいはLDPC符号などで用いられる従来のインターリーバでは、並べ替えのための専用のハードウェアを用いる場合と、並べ替えの情報をストアしたテーブルを用いる場合があるが、それぞれ非常に複雑な回路と大きな回路規模を必要とし、実現するためのコストが多大となっている。

【0016】また、並べ替え以後、パンクチャ、あるいは演算によりデータ数が減少する場合に、保持する必要のないデータを保持していることになり、回路規模の増大を招いている。

【0017】そこで、本発明の目的は、上述の如き従来

の問題点に鑑み、ターボ符号あるいはLDPC符号などで用いられるインターリーバにおいて必要とされるメモリを削減し、少ないハードウェア量にて確実に並べ替え処理を行うことができるインターリーブ処理方法及びインターリーブ処理装置を提供することにある。

【0018】

【課題を解決するための手段】本発明に係るインターリーブ処理方法は、入力されるデータシンボル系列あるいはデータビット系列に対し、インターリーブ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックとして、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる操作を行うことにより、入力される系列に対するインターリーブ操作の代わりとすることを特徴とする。

【0019】本発明に係るインターリーブ処理方法では、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる際に、過去に当該ブロックに割り当てられたデータ及び現在割り当てられようとしているデータを引数とするブロック単位で実施される所定の演算を行う。

【0020】また、本発明に係るインターリーブ処理方法では、ターボ符号あるいはこれに類する符号にて用いられるブロック符号において、ブロック符号の符号長を上記ブロックとし、ブロック符号の制約条件を上記ブロックに割り当てる演算を行う。

【0021】本発明に係るインターリーブ処理装置は、入力されるデータシンボル系列あるいはデータビット系列に対し、インターリーブ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックとして、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるための並べ替えデータが書き込まれた並べ替えテーブルと、上記並べ替えテーブルにより与えられる並べ替えデータに従って、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てるセレクタと、上記セレクタを介して上記データシンボル系列あるいはデータビット系列が割り当てる複数のメモリブロックとからなることを特徴とする。

【0022】本発明に係るインターリーブ処理装置において、上記複数のメモリブロックは、インターリーブ前のデータシンボル系列あるいはデータビット系列を所定のブロックに割り当てる際に、過去に当該ブロックに割り当てられたデータ及び現在割り当てられようとしているデータを引数とするブロック単位で実施される所定の演算を行う演算手段をそれぞれ備える。

【0023】また、本発明に係るインターリーブ処理装置では、ターボ符号あるいはこれに類する符号にて用いられるブロック符号において、ブロック符号の符号長を上記ブロックとし、ブロック符号の制約条件を上記ブロ

ックに割り当てる演算を上記演算手段にて行う。

【0024】

【発明の実施の形態】以下、この発明の実施の形態について図面を参照して詳細に説明する。

【0025】本発明は、ターボ符号あるいは、これに類する符号に用いられるインターリーブを用いたパリティ分散伝送を行っている信号処理装置の符号化部及び復号化部に含まれるインターリーバに適用される。

【0026】本発明に係るインターリーバでは、ターボ符号あるいは、これに類する符号などで用いられるインターリーバの代わりに、次に述べるblock accumulating interleaverを用いてデータ系列の変換を行う。

【0027】すなわち、本発明では、入力されるデータシンボル系列、あるいはデータビット系列に対し、インターリーブ操作を実施する場合、インターリーブ後の系列を所定のシンボル数あるいはビット数ごとに区切った部分系列をブロックと呼び、インターリーブ後の系列はブロックの系列として取り扱う。そして、インターリーブ前のデータシンボル系列、あるいはデータビット系列を所定のブロックに割り当てる操作（この操作をBlock Accumulating Interleavingと呼ぶ）を用いることで、入力される系列に対するインターリーブ操作の代わりとする。

【0028】ここで、4096ビットのデータ系列に対して並べ替えを行なうインターリーバを用いる場合を例とし、このとき、block accumulating interleaverを用いるインターリーバの構成の例を図1に示す。

【0029】この図1に示すインターリーバ10は、それぞれ1ビットのメモリMと併他の論理回路EXORで構成される64個のメモリブロックMB1～MB64と、入力データ系列を所定のブロックに割り当てるセレクタSLと、入力データ系列を所定のブロックに割り当てるための並べ替えデータが書き込まれた並べ替えテーブルTBからなる。

【0030】ここで、ブロック符号として64/65シングルパリティブロック符号では64ビットのデータに対し、全64ビットの併他の論理をとった結果の1ビットを加えた65ビットを符号語とする。ただし、ターボ符号、あるいはLDPC符号ではパリティである1ビットのみを伝送し、残りの64ビットのデータは所謂パンクチャすることにより伝送しない。4096ビットのデータ系列を所定の方法によりインターリーブした4096ビットの系列を仮定し、例えば64ビットごとの部分系列をブロックと呼ぶ。つまり、インターリーブ後の系列は1ブロック（64ビット）×64個の4096ビットで構成立っている。

【0031】次に、本発明を具体的な系列例を用いて説明する。

【0032】入力される4096ビットのデータ系列を

次のように定義する。

【0033】 $\{a_0, a_1, a_2, a_3, \dots, a_{4095}\}$ このデータ系列に対し、従来のインターリーブ手法を適用して並べ替えた結果を次のように定義する。

【0034】 $\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$ つまり、系列 $\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$ は、データ系列 $\{a_0, a_1, a_2, a_3, \dots, a_{4095}\}$ を所定の規則を用いて並べ替えたものである。所定の並べ替えを実現するためには次のような4096個のインデックスを持つ並べ替え回路が必要となる。

【0035】 a_0 が b_0 から b_{4096} のどのビットに並べ替えられるかを示すインデックス

a_1 が b_0 から b_{4096} のどのビットに並べ替えられるかを示すインデックス

a_2 が b_0 から b_{4096} のどのビットに並べ替えられるかを示すインデックス：

a_{4095} が b_0 から b_{4096} のどのビットに並べ替えられるかを示すインデックス

系列 $\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$ に対して、6/4/65シングルパリティ符号を適用する場合、当該系列の先頭より、64ビットごとにデータを区切り、次のような演算を実施する。

【0036】 $c_0 = (b_0 + b_1 + b_2 + b_3 + \dots + b_{63}) \bmod 2$

$c_1 = (b_{64} + b_{65} + b_{66} + \dots + b_{127}) \bmod 2$

:

$c_{63} = (b_{4092} + b_{4093} + \dots + b_{4095}) \bmod 2$

つまり、 $\{c_0, c_1, \dots, c_{63}\}$ は、 $\{b_0, b_1, \dots, b_{4095}\}$ を先頭より64ビットごとのブロックに区切り、それぞれのブロックに対して、2を法とする加算を実施、つまり全ビットの併他の論理と演算を実施したものの系列となる。4096ビットの入力データ系列 $\{a_0, a_1, a_2, a_3, \dots, a_{4095}\}$ に対して、64ビットのパリティ系列 $\{c_0, c_1, \dots, c_{63}\}$ を出力することになる。このとき、入力データ系列 $\{a_0, a_1, a_2, a_3, \dots, a_{4095}\}$ を、一時的に系列 $\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$ に並べ替えることなしに、出力のパリティ系列 $\{c_0, c_1, \dots, c_{63}\}$ を計算すれば、一時的な系列 $\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$ をストアするメモリ及び4096ビット並べ替えのためのセレクタの回路を削減することが可能となる。

【0037】そこで、本発明では次のようなグループ化を行っている。このグループ化が当発明におけるブロックである。一時的な系列

$\{b_0, b_1, b_2, b_3, \dots, b_{4095}\}$

を、 $64/65$ シングルパリティ符号を適用する 64 ビットごとにグルーピングすると次のようになる。

【0038】

{ { b₀, b₁, b₂, ..., b₆₃ },
 { b₆₄, b₆₅, ..., b₁₂₇ },
 ...
 { b₄₀₉₃₂, b₄₀₉₃₃, ..., b₄₀₉₄₅ } }

これらのグループに対して次のように名前を付ける。

【0039】 {group0, group1, ..., group63}
つまり、次のようにグルーピングしたことになる。

【0040】 Group0 = { b₀, b₁, ..., b₆₃ }
Group1 = { b₆₄, b₆₅, ..., b₁₂₇ }
:
Group63 = { b₄₀₉₃₂, b₄₀₉₃₃, ..., b₄₀₉₄₅ }

このとき、一時的な並べ替えを次のように行う。
【0041】 a₀ が Group0 から Group63 のどのブロックに並べ替えられるかを示すインデックス

a₁ が Group0 から Group63 のどのブロックに並べ替えられるかを示すインデックス
a₂ が Group0 から Group63 のどのブロックに並べ替えられるかを示すインデックス：
a₄₀₉₅ が Group0 から Group63 のどのブロックに並べ替えられるかを示すインデックス

さらに、各ブロックにおいて、上記 $64/65$ シングルパリティ符号を適用する。

【0042】 続いて、図 1 に示したインターリーバ 10 の動作の説明を行う。

【0043】 1 つのブロックに対して $64/65$ シングルパリティ符号を適用する場合、実際に伝送されるのは 1 ビットのパリティデータのみである。そこで、1 つのブロックに対して 1 ビットのメモリを割り当ればよい。入力される 4096 ビットのデータを 64 個のブロックのうち、どのブロックに割り当ればよいかを示すテーブルを用いて、連続的に入力されるデータを割り当てられたブロックに入力する。各メモリプロック MB 1 ~ MB 64 は、それぞれ 1 ビットのメモリ M と排他的論理和演算回路 EXOR で構成している。このとき、各メモリプロックでは、直前にストアされていたデータと、新たに入力されるデータとの排他的論理和をとり、再び 1 ビットのメモリ M にストアする。このような構成することで、従来、4096 ビット分のメモリと、並べ替えテーブル（1 ビット × 4096 個）を持つ必要があったが、本発明では 64 ビット分のメモリと、並べ替えテーブル（6 ビット × 4096 個）のみで実現可能である。

【0044】 従来は、入力データ系列を並べ替えた系列

を保持するためにメモリを必要としたが、本発明では伝送に必要とする符号系列を保持するためのメモリのみを必要とするために無駄な回路コストを削減することが可能となった。

【0045】 発明の変形例

上述の実施の形態では符号化回路についてのみ例をあげたが、復号時のインターリーバに対しても同様の手法を利用することができる。

【0046】 例えば、 $64/65$ シングルパリティ符号を用いた場合、4096 個の信頼度情報の入力データ列を仮定し、65 個のデータを 1 つのブロックにグルーピングしたとき、ブロック毎の演算を入力される信頼度情報を用いたブロックの信頼度を計算する演算とすることで、block accumulating interleaver を構成することが可能である。

【0047】 図 2 に復号器におけるインターリーバ 20 の構成を示す。

【0048】 この図 2 に示すインターリーバ 20 は、それぞれ 1 つの信頼度情報をストアするメモリと信頼度情報より信頼度を演算する回路で構成される 64 個のメモリプロック MB 1 ~ MB 64 と、受信データ系列を所定のブロックに割り当てるセレクタ S L と、受信データ系列を所定のブロックに割り当てるための並べ替えデータは書き込まれた並べ替えテーブル T B からなる。

【0049】 また、上述の実施の形態では $64/65$ シングルパリティ符号を用いたが、任意のブロック符号においても本発明を応用することができる。

【0050】

【発明の効果】 以上述べたように、本発明によれば、従来のインターリーバと比較して、少ないハードウェア量で同様の機能を有する回路を実現できる。

【図面の簡単な説明】

【図 1】 本発明に係る符号器におけるインターリーバの構成例を示すブロック図である。

【図 2】 復号器におけるインターリーバの構成を示すブロック図である。

【図 3】 ターボ符号の符号器の基本構成を示すブロック図である。

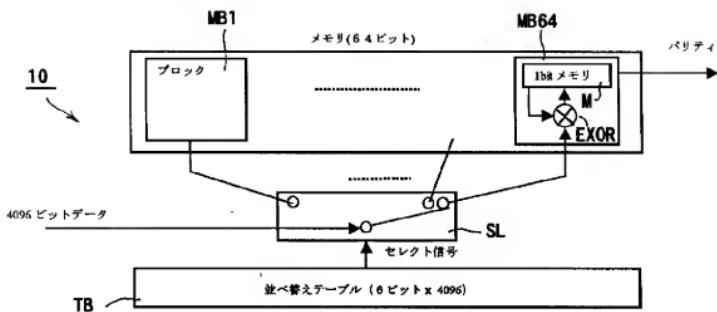
【図 4】 ターボ符号の復号器の基本構成を示すブロック図である。

【図 5】 4096 ビットのデータ系列に対して並べ替えを行なうインターリーバと $64/65$ シングルパリティ符号化回路の構成例を示すブロック図である。

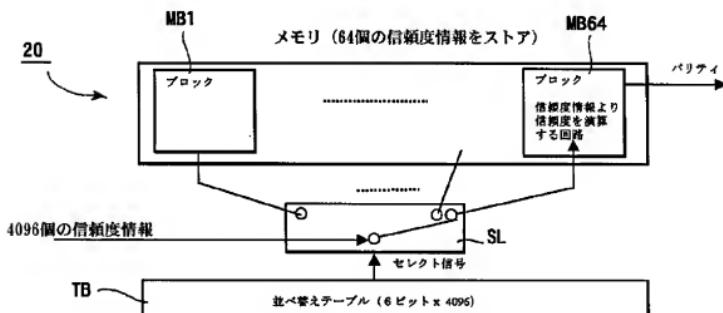
【符号の説明】

10, 20 インターリーバ、M メモリ M、EXOR 排他的論理和回路、MB 1 ~ MB 64 メモリプロック、S L セレクタ、T B 並べ替えテーブル

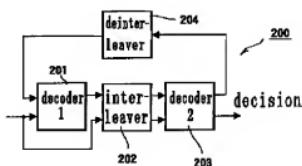
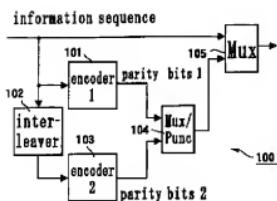
【図1】



【図2】



【図3】



【図5】

